UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/783,761	02/20/2004	Sungdo Moon	200313044-1	7412

22879 7590 07/27/2007
HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
WANG, BEN C	
ART UNIT	**PAPER NUMBER**
2192	

MAIL DATE	DELIVERY MODE
07/27/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/783,761	MOON ET AL.	
	Examiner	Art Unit	
	Ben C. Wang	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 May 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1 and 3-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 and 3-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Applicant's amendment dated May 7, 2007, responding to the Office action mailed January 22, 2007 provided in the rejection of claims 1-20, wherein claims 1 and 10 have been amended, claim 2 has been canceled, and claims 3-9 and 11-20 are remained as original.

Claims 1 and 3-20 remain pending in the application and which have been fully considered by the examiner.

Applicant's arguments with respect to claims rejection have been fully considered but are moot in view of the new grounds of rejection – see *Prakash et al.* art made of record, as applied hereto.

Claim Rejections – 35 USC § 103(a)

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1 and 3-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ayers et al. (*Aggressive Inlining, 1997, ACM*) (hereinafter 'Ayers') in view of Prakash et al., (Pub. No. US 2003/0005421 A1) (hereinafter 'Prakash' - art made of record)

3. **As to claim 1** (Currently Amended), Ayers discloses a method for cross-module in-lining (Sec. 1, 4th Para., Lines 7-9), comprising: In a first phase of a compiling process, the compiling process comprising a front-end phase, an inter-procedure analysis phase, the inter-procedure phase being the first phase, deciding to in-line a first function in a first module into a second function in a second module (Fig. 4 - screen inline candidates & select inline sites; Sec. 2.4, 3rd Para., Lines 1-2, 5-17); providing the location of the first function (Sec. 2.4, 1st Para, Lines 1-2; Sec. 2.3, 6th Para., Lines 7-13 – the address of a file-static procedure); providing instructions for in-lining to be performed in a second phase of the compiling process (Sec. 2.2, 1st Para., Lines 1-4, 6-9; Sec. 2.1, 1st Para., Lines 4-8).

Ayers discloses the High Level intermediate code Optimizer (HLO) which is analogous to applicant's Inter-procedural Analysis Phase (IPA) (Sec. 1, 4th Para.), but does not explicitly disclose in the second phase of the compiling process, the back-end phase being the second phase, following the instructions to in-line code of the first function into the second function.

However, in an analogous art of Inter-procedural Optimization Framework (IPO), Prakash discloses in the second phase of the compiling process, the back-end phase being the second phase, following the instructions to in-line code of the first function into the second function (Fig. 5 – a block diagram illustrating the use of all pre-IP files and an optimizer to create an output file; [0039] – in the second phase, the IPO extracts the "*ir*" information as a file from each of the object files and passes "*ir*" information to

optimizer in a single invocation step whereupon inter-procedural optimization is performed; Optimized "*ir*" files are passed to a code generator to generate code; Optimized object files are process by the IPO to generate Post-IPO object file; [0059] – PostIPO object files, due to cross module optimizations (i.e., cross target file optimization), have assumptions about other files; for example, if a PostIPO object file inlines a function "*foo*" from a first file, a second file assumes a certain definition of "*foo*".).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Prakash into the Ayers's system to further provide in the second phase of the compiling process, the back-end phase being the second phase, following the instructions to in-line code of the first function into the second function in Ayers system.

The motivation is that it would further enhance the Ayers's system by taking, advancing and/or incorporating Prakash's system which offers significant advantages for a method and a system that will make use of an optimizer that simultaneously interrelates and makes use of multiple program files in creating an efficient output file as once suggested by Prakash (e.g., [0011]).

4. **As to claim 10** (Currently Amended), Ayers discloses a method for compiling a first set of modules having programming source code (Fig 1 (bottom), element – Sources), comprising: In a first phase that represents a front-end phase (Fig. 1 (bottom), elements - Sources, Front End), from the first set of modules, providing a second set of

modules having first intermediate representations (Sec. 2.1, 1st Para., Lines 1-4); In a second phase that represents an inter-procedural phase, (Fig. 1 (bottom), element - Inter-procedural Optimization), performing in-line analysis on the second set of modules (Sec. 2.2, 1st Para., Lines 6-8); providing instructions for in-lining to be performed in a third phase of the compiling process (Sec. 2.2, 1st Para., Lines 9-12); and providing a third set of modules having second intermediate representations optimized (Sec. 1, 4th Para., Lines 1-3; Sec. 2.2, 1st Para., Lines 1-6, 2nd Para.) from the first intermediate representations (Sec. 2.2, 1st Para., Lines 9-12 - ucode); In the third phase of the compiling process.

Ayers discloses the High Level intermediate code Optimizer (HLO) which is analogous to applicant's Inter-procedural Analysis Phase (IPA) (Sec. 1, 4th Para.), but does not explicitly disclose the third phase representing a back-end phase, following the instructions to perform in-lining, and providing a fourth set of modules having third intermediate representations optimized from the second intermediate representations.

However, in an analogous art of Inter-procedural Optimization Framework (IPO), Prakash discloses the third phase representing a back-end phase, following the instructions to perform in-lining, and providing a fourth set of modules having third intermediate representations optimized from the second intermediate representations (Fig. 5 – a block diagram illustrating the use of all pre-IP files and an optimizer to create an output file; [0039] – in the second phase, the IPO extracts the "*ir*" information as a file from each of the object files and passes "*ir*" information to optimizer in a single invocation step whereupon inter-procedural optimization is performed; Optimized "*ir*"

files are passed to a code generator to generate code; Optimized object files are process by the IPO to generate Post-IPO object file; [0059] – PostIPO object files, due to cross module optimizations (i.e., cross target file optimization), have assumptions about other files; for example, if a PostIPO object file inlines a function "*foo*" from a first file, a second file assumes a certain definition of "*foo*".).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Prakash into the Ayers's system to further provide the third phase representing a back-end phase, following the instructions to perform in-lining, and providing a fourth set of modules having third intermediate representations optimized from the second intermediate representations in Ayers system.

The motivation is that it would further enhance the Ayers's system by taking, advancing and/or incorporating Prakash's system which offers significant advantages for a method and a system that will make use of an optimizer that simultaneously interrelates and makes use of multiple program files in creating an efficient output file as once suggested by Prakash (e.g., [0011]).

5. **As to claim 15** (original), Ayers discloses a computer-readable medium embodying a compiler, the compiler comprising: a front-end phase (Fig. 1 (bottom), element – Front End); a cross-module analysis phase (Fig. 1 (bottom), element – Inter-procedural Optimization); and a back-end phase (Fig. (bottom), element – Per-Routine

Optimization); wherein the front-end phase invokes the cross-module analysis phase (Sec. 2, 1st Para., Lines 1-4; Fig. 1 (bottom), element - Sources).

Ayers discloses the High Level intermediate code Optimizer (HLO) which is analogous to applicant's Inter-procedural Analysis Phase (IPA) (Sec. 1, 4th Para.), but does not explicitly disclose the cross-module analysis phase determines whether a callee is to be in-lined into a caller in the back-end phase; provides instructions for the back-end phase to transform in-lining code of the callee; and invokes the back-end phase; and the back-end phase transforms the in-lining code based on the instructions.

However, in an analogous art of Inter-procedural Optimization Framework (IPO), Prakash discloses the cross-module analysis phase determines whether a callee is to be in-lined into a caller in the back-end phase; provides instructions for the back-end phase to transform in-lining code of the callee; and invokes the back-end phase; and the back-end phase transforms the in-lining code based on the instructions (Fig. 5 – a block diagram illustrating the use of all pre-IP files and an optimizer to create an output file; [0039] – in the second phase, the IPO extracts the "*ir*" information as a file from each of the object files and passes "*ir*" information to optimizer in a single invocation step whereupon inter-procedural optimization is performed; Optimized "*ir*" files are passed to a code generator to generate code; Optimized object files are process by the IPO to generate Post-IPO object file; [0059] – PostIPO object files, due to cross module optimizations (i.e., cross target file optimization), have assumptions about other files; for example, if a PostIPO object file inlines a function "*foo*" from a first file, a second file assumes a certain definition of "*foo*".).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Prakash into the Ayers's system to further provide the cross-module analysis phase determines whether a callee is to be in-lined into a caller in the back-end phase; provides instructions for the back-end phase to transform in-lining code of the callee; and invokes the back-end phase; and the back-end phase transforms the in-lining code based on the instructions Ayers system.

The motivation is that it would further enhance the Ayers's system by taking, advancing and/or incorporating Prakash's system which offers significant advantages for a method and a system that will make use of an optimizer that simultaneously interrelates and makes use of multiple program files in creating an efficient output file as once suggested by Prakash (e.g., [0011]).

6. **As to claim 2** (canceled)

7. **As to claim 3** (original), Ayers discloses the method in the first phase of the compiling process, further having a third function in the module containing the second function (Sec. 2.4, 3rd Para., Lines 5-17).

8. **As to claim 4** (original), Ayers discloses the method in the second phase of the compiling process, further getting rid of the third function in the module containing the

second function after using that third function to in-line its code into the second function (Sec. 3.2, 3rd Para., Lines 6-10).

9. **As to claim 5** (original), Ayers discloses the method wherein the third function being selected from a group (Sec. 1, 4th Para., Lines 1-3) consisting of the first function (Sec. 1, 2nd Para., Lines 1-3) and a clone of the first function (Sec. 1, 2nd Para., Lines 7-11).

10. **As to claim 6** (original), Ayers discloses the method wherein in the second phase of the compiling process, in-lining the code of the first function into the second function uses a clone of the first function (Sec. 2.3; Fig. 3).

11. **As to claim 7** (original), Ayers discloses the method wherein in the second phase of the compiling process, the code used to be in-lined into the second function is stored in a file (Sec. 2.1, 1st Para., Lines 8-15).

12. **As to claim 8** (original), Ayers discloses the method wherein in the second phase of the compiling process, the code used to be in-lined into the second function is stored in a library (Sec. 3.1, 2nd Para., Lines 1-7).

13. **As to claim 9** (original), Ayers discloses the method wherein the instructions include at least a list of callees to be in-lined and corresponding callers (Sec. 1, 4th Para., Lines 7-13 – it can inline or clone calls both within and across program modules).

14. **As to claim 11** (original), Ayers discloses the method in the second phase, further using code in the module containing a function caller of a function callee to transform in-lining (Sec. 1, 2nd Para, Lines 1-3).

15. **As to claim 12** (original), Ayers discloses the method wherein the code being selected from a body of the function callee (Sec. 1, 2nd Para., Lines 1-3).

16. **As to claim 13** (original), Ayers discloses the method wherein the code being selected from a clone of the function callee (Sec. 1, 2nd Para., Lines 7-11).

17. **As to claim 14** (original), Ayers discloses the method wherein the instructions include at least one of: a set of function caller including at least one function caller (Fig. 4, Lines 3, 8); a set of function callee including at least one function callee (Sec. 2.4, 2nd Para., Line 1-2); the order for transformation of in-lining (Sec. 2.4, 2nd Para., Lines 2-3; 3rd Para., Lines 1-2); the location of at least one function callee (Sec. 2.3, 6th Para., Lines 7-12); and decisions whether to keep a body of at least one function callee after in-lining transformation (Sec. 3.2, 3rd Para., Lines 6-10).

18. **As to claim 16** (original), Ayers discloses the computer-readable medium wherein the back-end phase further performs tasks related to in-lining (Sec. 2.2, 3rd Para., Lines 8-11).

19. **As to claim 17** (original), Ayers discloses the computer-readable medium wherein the tasks related to in-lining include at least deleting the callee in a module containing the caller (Sec. 3.2, 3rd Para., Lines 6-10).

20. **As to claim 18** (original), Ayers discloses the computer readable medium wherein transforming the in-lining code uses code of a clone of the callee (Sec. 1, 2nd Para., Lines 7-11).

21. **As to claim 19** (original), Ayers discloses the computer-readable medium wherein a call to the callee is in a module that does not include the callee (Sec. 1, 4th Para., Lines 7-13).

22. **As to claim 20** (original), Ayers discloses the computer-readable medium wherein the instructions include at least a list of callees (Sec. 2.4, 2nd Para., Line 1-2).

Conclusion

23. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is 571-270-

1240. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

TUAN DAM
SUPERVISORY PATENT EXAMINER

BCW

July 16, 2007